

LAPORAN RESMI
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
OVERLOADING DAN OVERRIDING



Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TA 2022/2023

A. LATIHAN

1. Overriding

a. Class Base

```
class Base{
    private void amethod(int iBase){
        System.out.println("Base.amethod");
    }
}
```

b. Class Over

```
class Over extends Base{
    public static void main(String argv[]){
        Over o = new Over();
        int iBase=0;
        o.amethod(iBase);
    }

    public void amethod(int iOver){
        System.out.println("Over.amethod");
    }
}
```

c. Hasil kompilasi

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac Over.java
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java Over
Over.amethod
```

Analisa :

Pada program diatas, overriding method ditunjukkan pada baris 14, terdapat method dari subclass yang memiliki nama dan parameter yang sama dengan method superclass yaitu “amethod(int)”, sedangkan pada baris 2 merupakan overridden method.

2. Overloading

a. Class MyParent

```
class MyParent {
    int x, y;
    MyParent(int x, int y){
        this.x = x;
        this.y = y;
    }

    public int addMe(int x, int y){
        return this.x + x + y + this.y;
    }
}
```

```

    }

    public int addMe(MyParent myPar){
        return addMe(myPar.x, myPar.y);
    }
}

```

b. Class MyChild.java

```

class MyChild extends MyParent{
    int z;

    MyChild (int x, int y, int z) {
        super(x,y);
        this.z = z;
    }

    public int addMe(int x, int y, int z){
        return this.x + x + this.y + y + this.z + z;
    }

    public int addMe(MyChild myChi){
        return addMe(myChi.x, myChi.y, myChi.z);
    }

    public int addMe(int x, int y){
        return this.x + x + this.y + y;
    }
}

```

c. Class MySomeone.java

```

public class MySomeone{
    public static void main(String args[]) {
        MyChild myChi = new MyChild(10, 20, 30);
        MyParent myPar = new MyParent(10, 20);
        int x = myChi.addMe(10, 20, 30);
        int y = myChi.addMe(myChi);
        int z = myPar.addMe(myPar);
        System.out.println(x + y + z);
    }
}

```

d. Hasil compile

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac MySomeone.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java MySomeone
300

```

Analisa :

Pada class MyParent overloading method terletak pada baris ke 8 dan 12, sedangkan pada subclassnya, overloading method terletak pada baris ke 9 dan 13. Pada class MySomeOne baris ke 5 dan ke 6 merupakan pemanggilan method addme()

3. Overloading

a. Class MyClass

```
class MyClass{
    void myMethod(int i) {
        System.out.println("int version");
    }

    void myMethod(String s) {
        System.out.println("String version");
    }

    public static void main(String args[]){
        MyClass obj = new MyClass();
        char ch = 'c';
        obj.myMethod(ch);
    }
}
```

b. Hasil kompilasi

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac MyClass.java
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java MyClass
int version
```

Analisa :

Pada program di atas, terdapat 2 overloading method. Pada baris ke 13 dilakukan pemanggilan method.

4. Mengimplementasikan UML class diagram dalam program

a. Class Orang

```
public class Orang{
    protected String nama;
    protected int umur;

    public Orang(String nama){
        this.nama = nama;
    }

    public Orang(String nama, int umur){
        this.nama = nama;
        this.umur = umur;
    }
}
```

```
}  
}
```

b. Class Dosen

```
public class Dosen extends Orang{  
    private int nip;  
  
    public Dosen(String nama){  
        super(nama);  
    }  
  
    public Dosen(String nama, int nip){  
        super(nama);  
        this.nip = nip;  
    }  
  
    public Dosen(String nama, int nip, int umur){  
        super(nama, umur);  
        this.nip = nip;  
    }  
  
    public void Info(){  
        System.out.println("Nama : " + nama);  
        System.out.println("NIP : " + nip);  
        System.out.println("Umur " + umur);  
    }  
}
```

c. Class TesLatihan

```
public class TesLatihan{  
    public static void main(String args[]){  
        System.out.println("Memasukkan identitas dosen 1 : Agus");  
        Dosen dosen1 = new Dosen("Agus");  
        System.out.println("Memasukkan identitas dosen 2 : Budi,  
NIP. 1458");  
  
        Dosen dosen2 = new Dosen("Budi", 1458);  
        System.out.println("Memasukkan identitas dosen 3 : Iwan,  
NIP. 1215, umur 47");  
  
        Dosen dosen3 = new Dosen("Iwan", 1215, 47);  
  
        System.out.println();  
        dosen1.Info();  
        System.out.println();  
        dosen2.Info();  
        System.out.println();  
        dosen3.Info();  
    }  
}
```

```
}  
}
```

d. Hasil compile

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac TesLatihan.java  
  
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java TesLatihan  
Memasukkan identitas dosen 1 : Agus  
Memasukkan identitas dosen 2 : Budi, NIP. 1458  
Memasukkan identitas dosen 3 : Iwan, NIP. 1215, umur 47  
  
Nama : Agus  
NIP : 0  
Umur 0  
  
Nama : Budi  
NIP : 1458  
Umur 0  
  
Nama : Iwan  
NIP : 1215  
Umur 47
```

B. TUGAS

1. Mengimplementasikan UML class diagram dalam program

a. Class RerataNilai

```
public class RerataNilai{  
    public int average(int x, int y){  
        return (x+y) / 2;  
    }  
  
    public double average(double x, double y){  
        return (x+y) / 2;  
    }  
  
    public int average(int x, int y, int z){  
        return (x+y+z) / 3;  
    }  
}
```

b. Class TesTugas1

```
public class TesTugas1{  
    public static void main(String args[]){  
        RerataNilai rn = new RerataNilai();  
        System.out.println("Rerata nilai 21 dan 13  
adalah : " + rn.average(21, 13));  
    }  
}
```

```

        System.out.println("Rerata nilai 19.3 dan 39.5
adalah : " + rn.average(19.3, 39.5));
        System.out.println("Rerata nilai 123, 567 dan
744 adalah : " + rn.average(123, 567, 744));
    }
}

```

c. Hasil compile

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac TesTugas1.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java TesTugas1
Rerata nilai 21 dan 13 adalah : 17
Rerata nilai 19.3 dan 39.5 adalah : 29.4
Rerata nilai 123, 567 dan 744 adalah : 478

```

2. Mengimplementasikan UML class diagram dalam program

a. Class Katak

```

public class Katak{
    private int umur;
    private String nama;

    public Katak(int umur, String nama){
        this.umur = umur;
        this.nama = nama;
    }

    public String caraBergerak(){
        return "melompat";
    }

    public int getUmur(){
        return umur;
    }

    public String getNama(){
        return nama;
    }
}

```

b. Class Kecebong

```

public class Kecebong extends Katak{
    private double panjangEkor;

    public Kecebong(int umur, String nama, double
panjang){
        super(umur, nama);
    }
}

```

```

        this.panjangEkor = panjang;
    }

    public String caraBergerak(){
        return "berenang";
    }

    public double getPanjangEkor(){
        return panjangEkor;
    }
}

```

c. Class TesTugas2

```

public class TesTugas2{
    public static void main(String[] args){
        System.out.println("Objek \tumur \tnama
\t\tpanjangEkor \tcaraBergerak");

        System.out.println("=====
=====");

        Katak O1 = new Katak(5, "Froggy");
        System.out.printf("O1 \t%-7d %-15s %-15s
%s\n", O1.getUmur(), O1.getNama(), "", O1.caraBergerak());

        Kecebong O2 = new Kecebong(2, "Junior
Frog", 10);
        System.out.printf("O2 \t%-7d %-15s %-15.0f
%s\n", O2.getUmur(), O2.getNama(), O2.getPanjangEkor(),
O2.caraBergerak());

        System.out.println("=====
=====");

    }
}

```

d. Hasil kompile

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>javac TesTugas2.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>java TesTugas2

Objek   umur   nama           panjangEkor   caraBergerak
=====
O1      5      Froggy         melompat
O2      2      Junior Frog    10            berenang
=====

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 12>_

```